Terms used trace probe trap handler boot instrumented

Found 92 of 196,064

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 92 Result page: 1 2 3 4 5 next

Relevance scale

1 Trap-driven simulation with Tapeworm II

Richard Uhlig, David Nagle, Trevor Mudge, Stuart Sechrest

November 1994 **ACM SIGPLAN Notices , ACM SIGOPS Operating Systems Review , Proceedings of the sixth international conference on Architectural support for programming languages and operating systems ASPLOS-VI**, Volume 29 , 28 Issue 11 , 5

Publisher: ACM Press

Full text available: pdf(1.45 MB) Additional Information: full citation, abstract, references, citings, index terms

Tapeworm II is a software-based simulation tool that evaluates the cache and TLB performance of multiple-task and operating system intensive workloads. Tapeworm resides in an OS kernel and causes a host machine's hardware to drive simulations with kernel traps instead of with address traces, as is conventionally done. This allows Tapeworm to quickly and accurately capture complete memory referencing behavior with a limited degradation in overall system performance. This paper compares trap- ...

Keywords: TLB, cache, memory system, trace-driven simulation, trap-driven simulation

2 Virtual machines: ReVirt: enabling intrusion analysis through virtual-machine logging and replay

George W. Dunlap, Samuel T. King, Sukru Cinar, Murtaza A. Basrai, Peter M. Chen

December 2002 **ACM SIGOPS Operating Systems Review**, Volume 36 Issue SI

Publisher: ACM Press

Full text available: pdf(1.56 MB) Additional Information: full citation, abstract, references, citings

Current system loggers have two problems: they depend on the integrity of the operating system being logged, and they do not save sufficient information to replay and analyze attacks that include any non-deterministic events. ReVirt removes the dependency on the target operating system by moving it into a virtual machine and logging below the virtual machine. This allows ReVirt to replay the system's execution before, during, and after an intruder compromises the system, even if the intruder rep ...

3 Software-controlled caches in the VMP multiprocessor

D. R. Cheriton, G. A. Slavenburg, P. D. Boyle

June 1986 **ACM SIGARCH Computer Architecture News , Proceedings of the 13th annual international symposium on Computer architecture ISCA '86**, Volume 14 Issue 2

Publisher: IEEE Computer Society Press, ACM Press

Full text available: pdf(958.63 KB) Additional Information: full citation, abstract, references, citings, index terms

VMP is an experimental multiprocessor that follows the familiar basic design of multiple processors, each with a cache, connected by a shared bus to global memory. Each processor has a synchronous, virtually addressed, single master connection to its cache, providing very high memory bandwidth. An unusually large cache page size and fast sequential memory copy hardware make it feasible for cache misses to be handled in software, analogously to the handling of virtual memory page faults. Har ...

4 Threads and input/output in the synthesis kernal

H. Massalin, C. Pu

November 1989 **ACM SIGOPS Operating Systems Review , Proceedings of the twelfth ACM symposium on Operating systems principles SOSP '89**, Volume 23 Issue 5

Publisher: ACM Press

Full text available: pdf(1.34 MB) Additional Information: full citation, abstract, references, citings, index terms

The Synthesis operating system kernel combines several techniques to provide high performance, including kernel code synthesis, fine-grain scheduling, and optimistic synchronization. Kernel code synthesis reduces the execution path for frequently used kernel calls. Optimistic synchronization increases concurrency within the kernel. Their combination results in significant performance improvement over traditional operating system implementations. Using hardware and software emulating a SUN 3 ...

5 Fast detection of communication patterns in distributed executions

Thomas Kunz, Michiel F. H. Seuren

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research CASCON '97**

Publisher: IBM Press

Full text available: pdf(4.21 MB) Additional Information: full citation, abstract, references, index terms

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

6 Performance evaluation of a communication system for transputer-networks based on monitored event traces

C. W. Oehlrich, A. Quick

April 1991 **ACM SIGARCH Computer Architecture News , Proceedings of the 18th annual international symposium on Computer architecture ISCA '91**, Volume 19 Issue 3

Publisher: ACM Press

Full text available: pdf(881.18 KB) Additional Information: full citation, references, citings, index terms

7 Using the SimOS machine simulator to study complex computer systems

Mendel Rosenblum, Edouard Bugnion, Scott Devine, Stephen A. Herrod

January 1997 **ACM Transactions on Modeling and Computer Simulation (TOMACS)**, Volume 7 Issue 1

Publisher: ACM Press

Full text available: pdf(731.76 KB) Additional Information: full citation, references, citings, index terms, review

Keywords: computer architecture, computer simulation, computer system performance analysis, operating systems

8 Middleware performance analysis: Performance monitoring of java applications

M. Harkema, D. Quartel, B. M. M. Gijsen, R. D. van der Mei

July 2002 **Proceedings of the 3rd international workshop on Software and performance WOSP '02**

Publisher: ACM Press

Full text available: pdf(219.69 KB) Additional Information: full citation, abstract, references, citings, index terms

Over the past few years, Java has evolved into a mature platform for developing enterprise applications. A critical factor for the commercial success of these applications is end-to-end performance, e.g., in terms of response times, throughput and availability. This raises the need for the development, validation and analysis of performance models to predict performance metrics of interest. To develop and validate performance models, insight in the execution behavior of the application is essent ...

Keywords: performance measurement and monitoring of java applications

9 Implementation aspects of a SPARC V9 complete machine simulator

Bill Clarke, Adam Czezowski, Peter Strazdins

January 2002 **Australian Computer Science Communications , Proceedings of the twenty-fifth Australasian conference on Computer science - Volume 4 ACSC '02**, Volume 24 Issue 1

Publisher: Australian Computer Society, Inc., IEEE Computer Society Press

Full text available: pdf(1.33 MB) Additional Information: full citation, abstract, references, index terms

In this paper we present work in progress in the development of a complete machine simulator for the UltraSPARC, an implementation of the SPARC V9 architecture. The complexity of the UltraSPARC ISA presents many challenges in developing a reliable and yet reasonably efficient implementation of such a simulator. Our implementation includes a heavily object-oriented design for the simulator modules and infrastructure, caching of repeated computations for performance, adding an OS (system call) emu ...

Keywords: SMP, SPARC V9 ISA, UltraSPARC, complete machine simulator, execution-driven simulation, object-oriented design

10 A framework for reducing the cost of instrumented code

Matthew Arnold, Barbara G. Ryder

May 2001 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2001 conference on Programming language design and implementation PLDI '01**, Volume 36 Issue 5

Publisher: ACM Press

Full text available: pdf(1.78 MB) Additional Information: full citation, abstract, references, citings, index terms

Instrumenting code to collect profiling information can cause substantial execution overhead. This overhead makes instrumentation difficult to perform at runtime, often preventing many known *offline* feedback-directed optimizations from being used in online systems. This paper presents a general framework for performing *instrumentation sampling* to reduce the overhead of previously expensive instrumentation. The framework is simple and effective, using code-duplication and *coun* ...

11 Communication and consistency protocols: Detailed cache coherence characterization for OpenMP benchmarks

Jaydeep Marathe, Anita Nagarajan, Frank Mueller

June 2004 **Proceedings of the 18th annual international conference on Supercomputing ICS '04**

Publisher: ACM Press

Full text available: pdf(358.00 KB) Additional Information: full citation, abstract, references, citings, index terms

Past work on studying cache coherence in shared-memory symmetric multiprocessors (SMPs) concentrates on studying aggregate events, often from an architecture point of view. However, this approach provides insufficient information about the exact sources of inefficiencies in parallel applications. For SMPs in contemporary clusters, application performance is impacted by the pattern of shared memory usage, and it becomes essential to understand coherence behavior in terms of the application progra ...

Keywords: SMPs, cache analysis, coherence protocols, dynamic binary rewriting, program instrumentation

12 The flight recorder: an architectural aid for system monitoring

Michael M. Gorlick

December 1991 **ACM SIGPLAN Notices , Proceedings of the 1991 ACM/ONR workshop on Parallel and distributed debugging PADD '91**, Volume 26 Issue 12

Publisher: ACM Press

Full text available: pdf(944.95 KB) Additional Information: full citation, references, citings, index terms

13 Computability classes for enforcement mechanisms

Kevin W. Hamlen, Greg Morrisett, Fred B. Schneider

January 2006 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 28 Issue 1

Publisher: ACM Press

Full text available: pdf(337.62 KB) Additional Information: full citation, abstract, references, citings, index terms

A precise characterization of those security policies enforceable by program rewriting is given. This also exposes and rectifies problems in prior work, yielding a better characterization of those security policies enforceable by execution monitors as well as a taxonomy of enforceable security policies. Some but not all classes can be identified with known classes from computational complexity theory.

Keywords: Program rewriting, edit automata, execution monitoring, inlined reference monitoring, reference monitors, security automata

14 Profile-based pretenuring

Stephen M. Blackburn, Matthew Hertz, Kathryn S. Mckinley, J. Eliot B. Moss, Ting Yang

January 2007 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 29 Issue 1

Publisher: ACM Press

Full text available: pdf(2.57 MB) Additional Information: full citation, abstract, references, index terms

Pretenuring can reduce copying costs in garbage collectors by allocating long-lived objects into regions that the garbage collector will rarely, if ever, collect. We extend previous work on pretenuring as follows: (1) We produce pretenuring advice that is neutral with respect to the garbage collector algorithm and configuration. We thus can and do combine advice from different applications. We find for our benchmarks that predictions using object lifetimes at each allocation site in Java program ...

Keywords: Garbage collection, lifetime prediction, pretenuring, profiling

15 Service infastructure and network management: Architecture and techniques for diagnosing faults in IEEE 802.11 infrastructure networks

Atul Adya, Paramvir Bahl, Ranveer Chandra, Lili Qiu

September 2004 **Proceedings of the 10th annual international conference on Mobile computing and networking MobiCom '04**

Publisher: ACM Press

Full text available: pdf(303.82 KB) Additional Information: full citation, abstract, references, citings, index terms

The wide-scale deployment of IEEE 802.11 wireless networks has generated significant challenges for Information Technology (IT) departments in corporations. Users frequently complain about connectivity and performance problems, and network administrators are expected to diagnose these problems while managing corporate security and coverage. Their task is particularly difficult due to the unreliable nature of the wireless medium and a lack of intelligent diagnostic tools for determining the cause ...

Keywords: IEEE 802.11, disconnected clients, fault detection, fault diagnosis, infrastructure wireless networks, rogue APs

16 Using certes to infer client response time at the web server

David Olshefski, Jason Nieh, Dakshi Agrawal

February 2004 **ACM Transactions on Computer Systems (TOCS)**, Volume 22 Issue 1

Publisher: ACM Press

Full text available: pdf(2.30 MB) Additional Information: full citation, abstract, references, citings, index terms

As businesses continue to grow their World Wide Web presence, it is becoming increasingly vital for them to have quantitative measures of the mean client perceived response times of their web services. We present Certes (CliEnt Response Time Estimated by the Server), an online server-based mechanism that allows web servers to estimate mean client perceived response time, as if measured at the client. Certes is based on a model of TCP that quantifies the effect that connection drops have on mean ...

Keywords: Web server, client perceived response time

17 SPAM: a microcode based tool for tracing operating system events

Stephen W. Melvin, Yale N. Patt

December 1987 **Proceedings of the 20th annual workshop on Microprogramming MICRO 20**

Publisher: ACM Press

Full text available: pdf(405.55 KB) Additional Information: full citation, abstract, references, citings, index terms

We have developed a tool called SPAM (for System Performance Analysis using Microcode), based on microcode modifications to a VAX 8600, that traces operating system events as a side-effect to normal execution. This trace of interrupts, exceptions, system calls and context switches can then be processed to analyze operating system behavior for the purpose of debugging, tuning or development. SPAM allows measurements to be made on a fully operating UNIX system with little perturbation (typica ...

18 Binary translation and architecture convergence issues for IBM system/390

Michael Gschwind, Kemal Ebcioğlu, Erik Altman, Sumedh Sathaye

May 2000 **Proceedings of the 14th international conference on Supercomputing ICS '00**

Publisher: ACM Press

Full text available: pdf(1.44 MB) Additional Information: full citation, abstract, references, index terms

We describe the design issues in an implementation of the ESA/390 architecture based on binary translation to a very long instruction word (VLIW) processor. During binary translation, complex ESA/390 instructions are decomposed into instruction "primitives" which are then scheduled onto a wide-issue machine. The aim is to achieve high instruction level parallelism due to the increased scheduling and optimization opportunities

which can be exploited by binary translation software ...

19 A microcode-based environment for noninvasive performance analysis

S. W. Melvin, Y. N. Patt

December 1986 **ACM SIGMICRO Newsletter , Proceedings of the 19th annual workshop on Microprogramming MICRO 19**, Volume 17 Issue 4

Publisher: ACM Press

Full text available: pdf(757.28 KB) Additional Information: full citation, abstract, references, citings, index terms

We have developed an environment which allows us to collect data for performance analysis by modifying the microcode of a VAX 8600. This use of microprogramming permits data to be collected with minimal system perturbation (i.e. the data is almost as good as that obtained with a hardware monitor) but at the cost and with the ease of use of a software simulator. In this paper we describe the environment that we have developed and present two examples of its use. The first example, procedure ...

20 Networked embedded software: S²DB: a novel simulation-based debugger for sensor network applications

Ye Wen, Rich Wolski, Selim Gurun

October 2006 **Proceedings of the 6th ACM & IEEE International conference on Embedded software EMSOFT '06**

Publisher: ACM Press

Full text available: pdf(221.61 KB) Additional Information: full citation, abstract, references, index terms

Sensor network computing can be characterized as resource-constrained distributed computing using unreliable, low bandwidth communication. This combination of characteristics poses significant software development and maintenance challenges. Effective and efficient debugging tools for sensor network are thus critical. Existent development tools, such as TOSSIM, EmStar, ATEMU and Avrora, provide useful debugging support, but not with the fidelity, scale and functionality that we believe are suffi ...

Keywords: ebugging, sensor network, simulation